



Housing Project

Submitted by
Alfinson John

ACKNOWLEDGMENT

In this project different libraries and methods are used that are available in python which helped in completion of the project:

<https://scikitlearn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html>

<http://scikitlearn.org/stable/modules/generated/sklearn.preprocessing.PowerTransformer.html>

<http://scikitlearn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html>

http://scikitlearn.org/stable/modules/generated/sklearn.model_selection.train_test_split.html

http://scikitlearn.org/stable/modules/generated/sklearn.metrics.model_evaluation.html

http://scikitlearn.org/stable/auto_examples/classification/plot_classifier_comparison.html

http://scikitlearn.org/stable/auto_examples/classification/plot_classifier_comparison.html

<https://seaborn.pydata.org/generated/seaborn.countplot.html>

<https://www.analyticsvidhya.com/blog/2020/06/roc-curve-machinelearning/>

<https://www.analyticsvidhya.com/blog/2020/10/how-to-choose-evaluation-metrics-for-classification-model/>

<https://machinelearningmastery.com/smote-oversampling-for-imbalanced-classification/>

INTRODUCTION

● Business Problem Framing Houses are one of the necessary need of each and every person around the globe and therefore housing and real estate market is one of the markets which is one of the major contributors in the world's economy. It is a very large market and there are various companies working in the domain. Data science comes as a very important tool to solve problems in the domain to help the companies increase their overall revenue, profits, improving their marketing strategies and focusing on changing trends in house sales and purchases. Predictive modelling, Market mix modelling, recommendation systems are some of the machine learning techniques used for achieving the business goals for housing companies. Our problem is related to one such housing company. A US-based housing company named Surprise Housing has decided to enter the Australian market. The company uses data analytics to purchase houses at a price below their actual values and flip them at a higher price. For the same purpose, the company has collected a data set from the sale of houses in Australia. The data is provided in the CSV file below. The company is looking at prospective properties to buy houses to enter the market. You are required to build a model using Machine Learning in order to predict the actual value of the prospective properties and decide whether to invest in them or not. For this company wants to know:

- Which variables are important to predict the price of variable?
- How do these variables describe the price of the house?
- Conceptual Background of the Domain Problem The project will require knowledge and practice in building Graphs /plots and analysing them to get the relationship between dataset, Knowledge of Different Learning Models to build and predict the required output. Basic Data

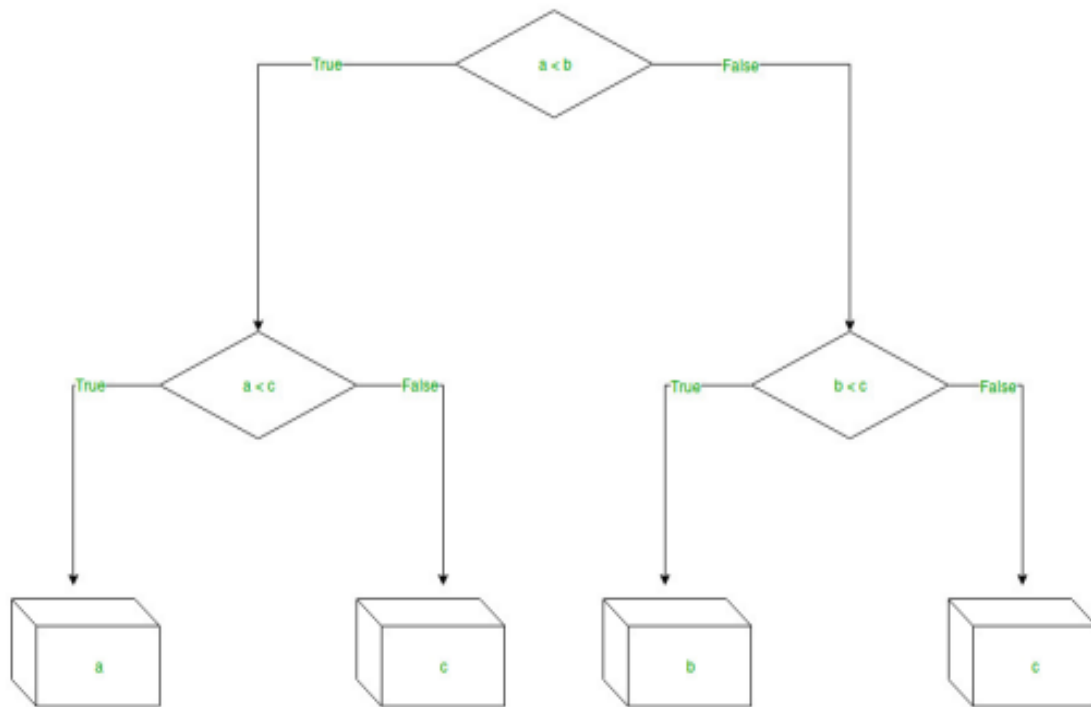
science concepts to increase the quality of the dataset and Python Knowledge (Coding Language) which will be used to solve the complete Micro Credit Defaulter project. Understanding of calculating F2 score, accuracy, skewness and basic mathematics/statistical approaches will help to build an accurate model for this project.

- Review of Literature Market price is what a willing, ready and bank-qualified buyer will pay for a property and what the seller will accept for it. The transaction that takes place determines the market price, which will then influence the market value of future sales. Price is determined by local supply and demand, the property's condition and what other similar properties have sold for without adding in the value component. Market value is an opinion of what a property would sell for in a competitive market based on the features and benefits of that property (the value), the overall real estate market, supply and demand, and what other similar properties have sold for in the same condition. The major difference between market value and market price is that the market value, in the eyes of the seller, might be much more than what a buyer will pay for the property or its true market price. Value can create demand, which can influence price. But, without the demand function, value alone cannot influence price. As supply increases and demand decreases, price goes down, and value is not influential. As supply decreases and demand increases, the price will rise, and value will influence price. Market value and market price can be equal in a balanced market. However, buyers and sellers can view value differently. A seller might feel that their in-ground pool is a benefit, but the buyer could see it as a negative and place less value on the property. Or the seller could feel the new roof they put on the house has great value; however, the buyer places no value on this because they expect the property to have a roof in good condition. Or a builder might feel he has superior quality and demand a higher price, but the buyer places less

value on quality and more value on the lot, neighborhood and floor plan of the property.

- Motivation for the Problem Undertaken I wanted to solve the real-life problem using the Technical skills gathered during the course of being a Data Analyst and improving the skill set. Analytical Problem Framing

- Mathematical/ Analytical Modelling of the Problem---- Regression Models-> Regression analysis is a form of predictive modelling technique which investigates the relationship between a dependent (target) and independent variable (s) (predictor). This technique is used for forecasting, time series modelling and finding the causal effect relationship between the variables. For example, relationship between rash driving and number of road accidents by a driver is best studied through regression. Decision Tree – It is a decision-making tool that uses a flowchart-like tree structure or is a model of decisions and all of their possible results, including outcomes, input costs and utility. Decision-tree algorithm falls under the category of supervised learning algorithms. It works for both continuous as well as categorical output variables. The branches/edges represent the result of the node and the nodes have either: Conditions [Decision Nodes] Result [End Nodes] The branches/edges represent the truth/falsity of the statement and takes makes a decision based on that in the example below which shows a decision tree that evaluates the smallest of three numbers:



Random Forest

Random forest is a supervised learning algorithm which is used for both classification as well as regression. But however, it is mainly used for classification problems.

As we know that a forest is made up of trees and more trees means more robust forest. Similarly, random forest algorithm creates decision trees on data samples and then gets the prediction from each of them and finally selects the best solution by means of voting. It is an ensemble method which is better than a single decision tree because it reduces the over-fitting by averaging the result. Naive Bayes – Naive Bayes classifiers are a collection of classification algorithms based on Bayes' Theorem. It is not a single algorithm but a family of algorithms where all of them share a common principle, i.e. every pair of features being classified is independent of each other. Linear Regression – Linear regression is a linear model, e.g. a model that assumes a linear relationship between the input variables (x) and the single output

variable (y). More specifically, that y can be calculated from a linear combination of the input variables (x).

When there is a single input variable (x), the method is referred to as simple linear regression. When there are multiple input variables, literature from statistics often refers to the method as multiple linear regression. SVM – Supervised Machine Learning Algorithm used for classification and/or regression. It is more preferred for classification but is sometimes very useful for regression as well. Basically, SVM finds a hyper-plane that creates a boundary between the types of data. In 2-dimensional space, this hyper-plane is nothing but a line. We used different Plots/ graphs to perform EDA on the dataset->

1) Box Plot: It is a type of chart that depicts a group of numerical data through their quartiles. It is a simple way to visualize the shape of our data. It makes comparing characteristics of data between categories very easy.

2) Count Plot: IT is kind of like a histogram or a bar graph for some categorical area. It simply shows the number of occurrences of an item based on a certain type of category

3) Heat Map: It contains values representing various shades of the same colour for each value to be plotted. Usually the darker shades of the chart represent higher values than the lighter shade. For a very different value a completely different colour can also be used.

4) Scatter Plot: A scatter plot is a diagram where each value in the data set is represented by a dot. The Matplotlib module has a method for drawing scatter plots

- Data Sources and their formats Below are the fields present in our dataset with the information what these fields describe MSSubClass: Identifies the type of dwelling involved in the sale. 20 1-STORY 1946 &

NEWER ALL STYLES 30 1-STORY 1945 & OLDER 40 1-STORY W/FINISHED ATTIC ALL AGES 45 1-1/2 STORY - UNFINISHED ALL AGES 50 1-1/2 STORY FINISHED ALL AGES 60 2-STORY 1946 & NEWER 70 2-STORY 1945 & OLDER 75 2-1/2 STORY ALL AGES 80 SPLIT OR MULTI-LEVEL 85 SPLIT FOYER 90 DUPLEX - ALL STYLES AND AGES 120 1-STORY PUD (Planned Unit Development) - 1946 & NEWER 150 1-1/2 STORY PUD - ALL AGES 160 2-STORY PUD - 1946 & NEWER 180 PUD - MULTILEVEL - INCL SPLIT LEV/FOYER 190 2 FAMILY CONVERSION - ALL STYLES AND AGES

MSZoning: Identifies the general zoning classification of the sale. A Agriculture C Commercial FV Floating Village Residential I Industrial RH Residential High Density RL Residential Low Density RP Residential Low Density Park RM Residential Medium Density

LotFrontage: Linear feet of street connected to property

LotArea: Lot size in square feet Street: Type of road access to property Grvl Gravel Pave Paved Alley: Type of alley access to property Grvl Gravel Pave Paved NA No alley access

LotShape: General shape of property Reg Regular IR1 Slightly irregular IR2 Moderately Irregular IR3 Irregular

LandContour: Flatness of the property Lvl Near Flat/Level Bnk Banked - Quick and significant rise from street grade to building HLS Hillside - Significant slope from side to side Low Depression Utilities: Type of utilities available AllPub All public Utilities (E,G,W,& S) NoSewr Electricity, Gas, and Water (Septic Tank) NoSeWa Electricity and Gas Only ELO Electricity only LotConfig: Lot configuration Inside Inside lot CornerCorner lot CulDSac Cul-de-sac FR2 Frontage on 2 sides of property FR3 Frontage on 3 sides of property

LandSlope: Slope of property Gtl Gentle slope Mod Moderate Slope Sev Severe Slope Neighborhood: Physical locations within Ames city

limits Blmngtn Bloomington Heights Blueste Bluestem BrDale Briardale
BrkSide Brookside ClearCr Clear Creek CollgCr College Creek Crawfor
Crawford Edwards Edwards GilbertGilbert IDOTRR Iowa DOT and Rail
Road MeadowV Meadow Village Mitchel Mitchell NamesNorth Ames
NoRidge Northridge NPKvill Northpark Villa NridgHt Northridge Heights
NWAmes Northwest Ames OldTown Old Town SWISU South & West of
Iowa State University Sawyer Sawyer SawyerW Sawyer West Somerst
Somerset StoneBr Stone Brook Timber Timberland Veenker Veenker

Condition1: Proximity to various conditions Artery Adjacent to arterial
street Feedr Adjacent to feeder street Norm Normal RRNn Within 200'
of North-South Railroad RRAn Adjacent to North-South Railroad PosN
Near positive off-site feature--park, greenbelt, etc. PosA Adjacent to
postive off-site feature RRNe Within 200' of East-West Railroad RRAe
Adjacent to East-West Railroad Condition2: Proximity to various
conditions (if more than one is present) Artery Adjacent to arterial
street Feedr Adjacent to feeder street Norm Normal RRNn Within 200'
of North-South Railroad RRAn Adjacent to North-South Railroad PosN
Near positive off-site feature--park, greenbelt, etc. PosA Adjacent to
postive off-site feature RRNe Within 200' of East-West Railroad RRAe
Adjacent to East-West Railroad BldgType: Type of dwelling 1Fam Single-
family Detached 2FmCon Two-family Conversion; originally built as one-
family dwelling Duplx Duplex Twnhse Townhouse End Unit
TwnhslTownhouse Inside Unit HouseStyle: Style of dwelling 1Story One
story 1.5Fin One and one-half story:

2nd level finished 1.5UnfOne and one-half story: 2nd level unfinished
2Story Two story 2.5Fin Two and one-half story: 2nd level finished
2.5Unf Two and one-half story: 2nd level unfinished SFoyer Split Foyer
SLvl Split Level OverallQual: Rates the overall material and finish of the
house 10 Very Excellent 9 Excellent 8 Very Good 7 Good 6 Above
Average 5 Average 4 Below Average 3 Fair 2 Poor 1 Very Poor

OverallCond: Rates the overall condition of the house 10 Very Excellent
9 Excellent 8 Very Good 7 Good 6 Above Average 5 Average 4 Below
Average 3 Fair 2 Poor 1 Very Poor YearBuilt:

Original construction date YearRemodAdd: Remodel date (same as
construction date if no remodeling or additions) RoofStyle: Type of roof
Flat Flat Gable Gable Gambrel Gabrel (Barn) Hip Hip Mansard Mansard
Shed Shed RoofMatl: Roof material ClyTile Clay or Tile CompShg
Standard (Composite) Shingle Membran Membrane Metal Metal Roll
Roll Tar&Grv Gravel & Tar WdShake Wood Shakes WdShngl Wood
Shingles Exterior1st:

Exterior covering on house AsbShng Asbestos Shingles AsphShn Asphalt
Shingles BrkComm Brick Common BrkFace Brick Face CBlock Cinder
Block CemntBd Cement Board HdBoard Hard Board ImStucc Imitation
Stucco MetalSd Metal Siding Other Other Plywood Plywood PreCast
PreCast Stone Stone Stucco Stucco VinylSd Vinyl Siding Wd Sdng Wood
Siding WdShing Wood Shingles Exterior2nd: Exterior covering on house
(if more than one material)

AsbShng Asbestos Shingles AsphShn Asphalt Shingles BrkComm Brick
Common BrkFace Brick Face CBlock Cinder Block CemntBd Cement
Board HdBoard Hard Board ImStucc Imitation Stucco MetalSd Metal
Siding Other Other Plywood Plywood PreCast PreCast Stone Stone
Stucco Stucco VinylSd Vinyl Siding Wd Sdng Wood Siding WdShing
Wood Shingles MasVnrType: Masonry veneer type BrkCmn Brick
Common BrkFace Brick Face CBlock Cinder Block None None Stone
Stone MasVnrArea:

Masonry veneer area in square feet ExterQual: Evaluates the quality of
the material on the exterior Ex Excellent Gd Good TA Average/Typical
Fa Fair Po Poor ExterCond: Evaluates the present condition of the
material on the exterior Ex Excellent Gd Good TA Average/Typical Fa

Fair Po Poor Foundation: Type of foundation BrkTil Brick & Tile CBlock Cinder Block PConc Poured Contrete Slab Slab Stone Stone Wood Wood BsmtQual: Evaluates the height of the basement Ex Excellent (100+ inches) Gd Good (90-99 inches) TA Typical (80-89 inches) Fa Fair (70-79 inches)

● Data Pre-processing Done

1) First we checked the data set dimensions

```
In [4]: df.shape
Out[4]: (1168, 81)
```

We have 1168 rows and 81 columns

2) Then we checked whether there is any repeating data available

```
duplicate = df.duplicated()
print(duplicate.sum())
df[duplicate]
```

0

3) We checked the outliers using the Box Plot and replaced the outliers with more appropriate values. Removal of outliers can also be done but taking the Data Loss percentage into consideration It is better to replace the outlier

● Hardware and Software Requirements and Tools Used

1) Software: Jupyter Notebook - To code and build the project in python

2) Libraries:

a) numpy - To perform basic math operations

- b) pandas - To perform basic File operations
- c) Matplotlib - To plot Different Graphs/ Plots
- d) Seaborn - Advance library to enhance the quality of graphs/plots
- e) warnings - To ignore the unwanted warnings raised while interpreting the code
- f) sklearn - To build the Prediction models
- g) imblearn - To balance our dataset distribution

Model/s Development and Evaluation

- Identification of possible problem-solving approaches (methods) We used different approaches from checking the dataset quality to building the model. We checked the null values and repeated rows in the dataset. For checking the Outliers, we used Box Plot and to remove the outliers we used IQR method. Then we moved to next step of checking data distribution and skewness. To scale the data, we used MinMax Scaler method and to remove the skewness we first checked the log and square root method but skewness of the dataset was not getting removed from it so we performed the Power transform to remove skewness.. We started building different models and checked their R2 score and selected the best suited model to perform Hyper tuning on. We got Random Forest Algo with the best result and after performing Hyper tuning we finalized the model.

- Testing of Identified Approaches (Algorithms)

- 1) Linear Regression
- 2) Decision Tree
- 3) Elastic Net

4) Lasso

5) Random Forest

6) Ridge

- Run and Evaluate selected models

```
In [36]: from sklearn import metrics
         regr = LinearRegression()
         regr.fit(x_train, y_train)
         pred=regr.predict(x_test)
         print('R2 score',r2_score(y_test, pred))
         print('MAE:', metrics.mean_absolute_error(y_test, pred))
         print('MSE:', metrics.mean_squared_error(y_test, pred))
         print('RMSE:', np.sqrt(metrics.mean_squared_error(y_test, pred)))
```

```
R2 score 0.9122522295050699
MAE: 14889.955766381772
MSE: 371736028.1239309
RMSE: 19280.45715547043
```

```
In [37]: rr = Ridge(alpha=0.01)
         rr.fit(x_train, y_train)
         pred=rr.predict(x_test)
         print('R2 score',r2_score(y_test, pred))
         print('MAE:', metrics.mean_absolute_error(y_test, pred))
         print('MSE:', metrics.mean_squared_error(y_test, pred))
         print('RMSE:', np.sqrt(metrics.mean_squared_error(y_test, pred)))
```

```
R2 score 0.911823459114965
MAE: 14934.890606094272
MSE: 373552477.7145672
RMSE: 19327.505729259716
```

```
In [38]: model_lasso = Lasso(alpha=0.01)
         model_lasso.fit(x_train, y_train)
         pred=model_lasso.predict(x_test)
         print('R2 score',r2_score(y_test, pred))
         print('MAE:', metrics.mean_absolute_error(y_test, pred))
         print('MSE:', metrics.mean_squared_error(y_test, pred))
         print('RMSE:', np.sqrt(metrics.mean_squared_error(y_test, pred)))
```

```
R2 score 0.9118234658159894
MAE: 14935.104867786691
MSE: 373552449.3262428
RMSE: 19327.504994857532
```

```
In [39]: model_enet = ElasticNet(alpha = 0.01)
model_enet.fit(x_train, y_train)
pred=model_enet.predict(x_test)
print('R2 score',r2_score(y_test, pred))
print('MAE:', metrics.mean_absolute_error(y_test, pred))
print('MSE:', metrics.mean_squared_error(y_test, pred))
print('RMSE:', np.sqrt(metrics.mean_squared_error(y_test, pred)))
```

```
R2 score 0.9119966779517421
MAE: 14928.021981156287
MSE: 372818650.72367525
RMSE: 19308.512390230255
```

```
In [40]: from sklearn.tree import DecisionTreeRegressor
from sklearn import metrics
dtr=DecisionTreeRegressor()
dtr.fit(x_train,y_train)
pred=dtr.predict(x_test)
print('R2 score', r2_score(y_test, pred))
print('MAE:', metrics.mean_absolute_error(y_test, pred))
print('MSE:', metrics.mean_squared_error(y_test, pred))
print('RMSE:', np.sqrt(metrics.mean_squared_error(y_test, pred)))
```

```
R2 score 0.77230316602535
MAE: 23381.713675213676
MSE: 964618430.767094
RMSE: 31058.30695268327
```

```
In [41]: from sklearn.ensemble import RandomForestRegressor
rdr = RandomForestRegressor()
rdr.fit(x_train,y_train)
pred1=rdr.predict(x_test)
print('R2 score',r2_score(y_test, pred1))
print('MAE:', metrics.mean_absolute_error(y_test, pred1))
print('MSE:', metrics.mean_squared_error(y_test, pred1))
print('RMSE:', np.sqrt(metrics.mean_squared_error(y_test,pred1)))
```

```
R2 score 0.8870777493514164
MAE: 15944.455769230768
MSE: 478385589.8121632
RMSE: 21872.027565183875
```

- Key Metrics for success in solving problem under consideration 1)
Mean Absolute Error(MAE) MAE is a very simple metric which calculates the absolute difference between actual and predicted values. To better understand, let's take an example you have input data and output data and use Linear Regression, which draws a best-fit line. Now you have to find the MAE of your model which is basically a mistake made by the model known as an error.

The diagram shows the formula for Mean Absolute Error (MAE):
$$MAE = \frac{1}{N} \sum |Y - \hat{Y}|$$
 Annotations include: 'Divide by total Number of Data Points' pointing to the $\frac{1}{N}$ term; 'Actual Output' pointing to Y and 'Predicted Output' pointing to \hat{Y} in the absolute value term; and 'Sum Of Absolute Value of residual' pointing to the summation symbol \sum and the absolute value term $|Y - \hat{Y}|$.

Now find the difference between the actual value and predicted value that is an absolute error but we have to find the mean absolute of the complete dataset. so, sum all the errors and divide them by a total number of observations And this is MAE.

And we aim to get a minimum MAE because this is a loss. Advantages of MAE The MAE you get is in the same unit as the output variable. It is most Robust to outliers. Disadvantages of MAE The graph of MAE is not differentiable so we have to apply various optimizers like Gradient descent which can be differentiable.

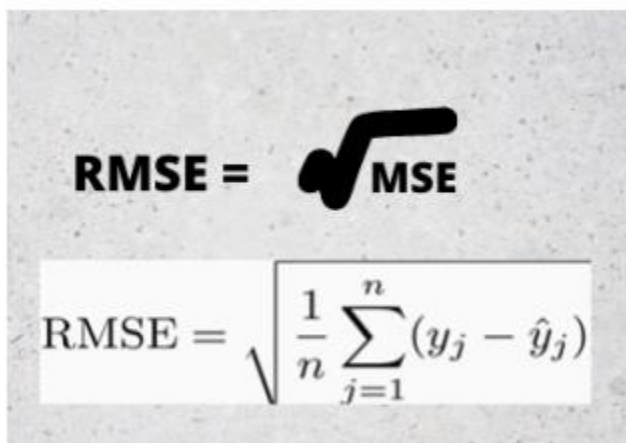
```
from sklearn.metrics import mean_absolute_error  
print("MAE",mean_absolute_error(y_test,y_pred))
```

Now to overcome the disadvantage of MAE next metric came as MSE

. 2) Mean Squared Error(MSE) MSE is a most used and very simple metric with a little bit of change in mean absolute error. Mean squared error states that finding the squared difference between actual and predicted value.

$$MSE = \frac{1}{n} \sum \left(\underbrace{y - \hat{y}}_{\substack{\text{The square of the difference} \\ \text{between actual and} \\ \text{predicted}}} \right)^2$$

So, above we are finding the absolute difference and here we are finding the squared difference. What actually the MSE represents? It represents the squared distance between actual and predicted values. we perform squared to avoid the cancellation of negative terms and it is the benefit of MSE. Advantages of MSE The graph of MSE is differentiable, so you can easily use it as a loss function. Disadvantages of MSE The value you get after calculating MSE is a squared unit of output. for example, the output variable is in meter(m) then after calculating MSE the output we get is in meter squared. If you have outliers in the dataset then it penalizes the outliers most and the calculated MSE is bigger.


$$RMSE = \sqrt{MSE}$$
$$RMSE = \sqrt{\frac{1}{n} \sum_{j=1}^n (y_j - \hat{y}_j)^2}$$

So, in short, It is not Robust to outliers which were an advantage in MAE. from sklearn.metrics import mean_squared_error
print("MSE",mean_squared_error(y_test,y_pred))

$$\mathbf{R^2\ Squared = 1 - \frac{SSr}{SSm}}$$

SSr = Squared sum error of regression line

SSm = Squared sum error of mean line

3) Root Mean Squared Error(RMSE) As RMSE is clear by the name itself, that it is a simple square root of mean squared error. Advantages of RMSE The output value you get is in the same unit as the required output variable which makes interpretation of loss easy. Disadvantages of RMSE It is not that robust to outliers as compared to MAE.

for performing RMSE we have to NumPy NumPy square root function over MSE. print("RMSE",np.sqrt(mean_squared_error(y_test,y_pred))) Most of the time people use RMSE as an evaluation metric and mostly when you are working with deep learning techniques the most preferred metric is RMSE.

4) Root Mean Squared Log Error(RMSLE) Taking the log of the RMSE metric slows down the scale of error. The metric is very helpful when you are developing a model without calling the inputs. In that case, the output will vary on a large scale. To control this situation of RMSE we take the log of calculated RMSE error and resultant we get as RMSLE. To perform RMSLE we have to use the NumPy log function over RMSE. print("RMSE",np.log(np.sqrt(mean_squared_error(y_test,y_pred))))) It is a very simple metric that is used by most of the datasets hosted for Machine Learning competitions.

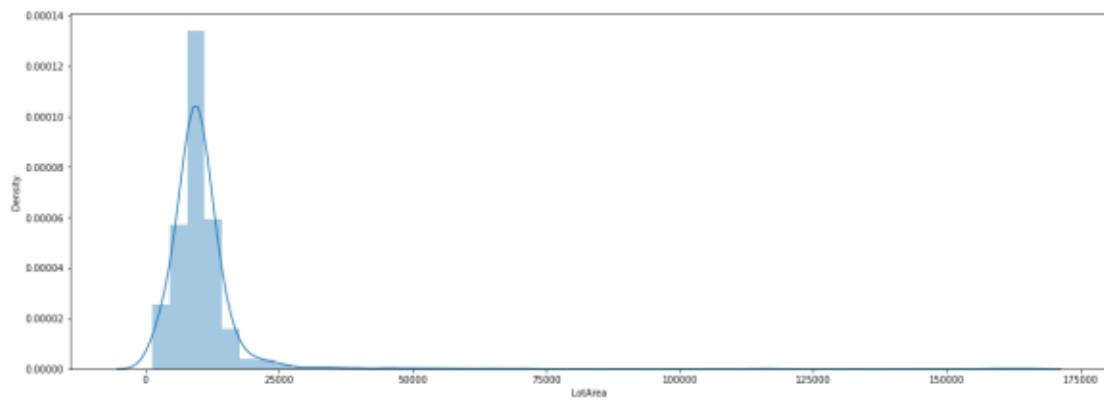
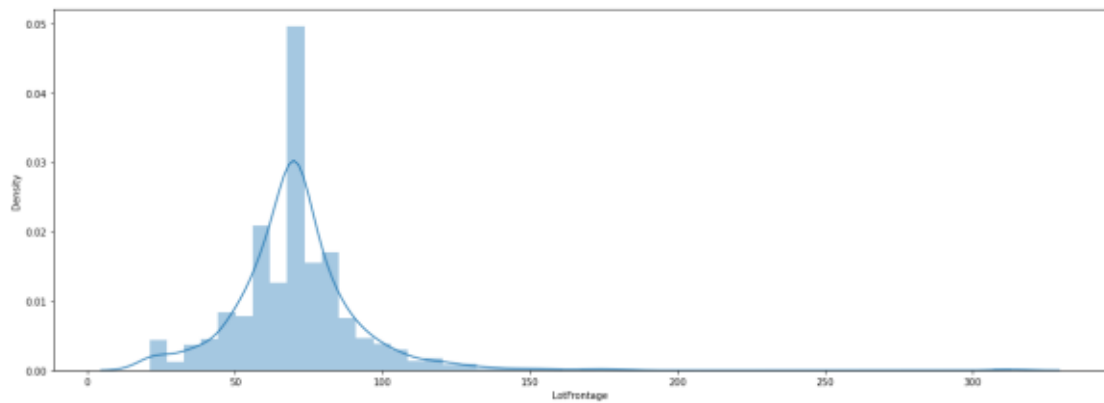
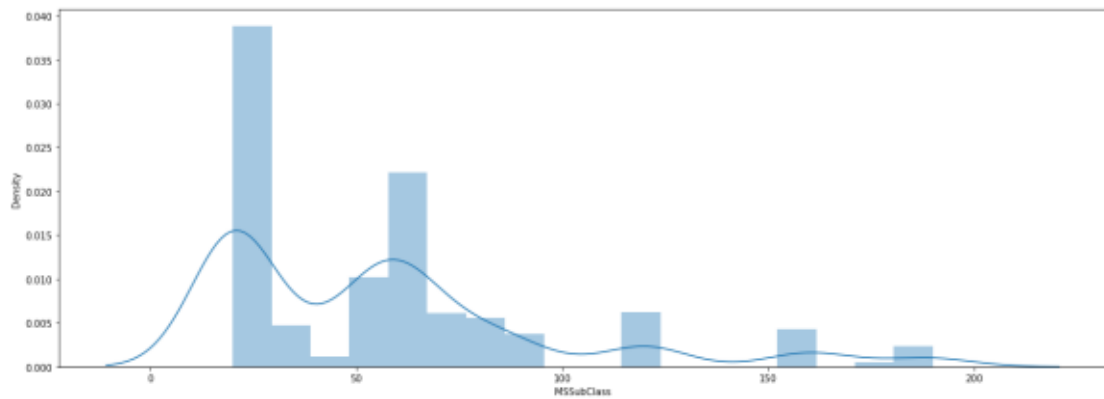
5) R Squared (R^2) R^2 score is a metric that tells the performance of your model, not the loss in an absolute sense that how many wells did your model perform. In contrast, MAE and MSE depend on the context as we have seen whereas the R^2 score is independent of context. So, with help of R squared we have a baseline model to compare a model which none of the other metrics provides.

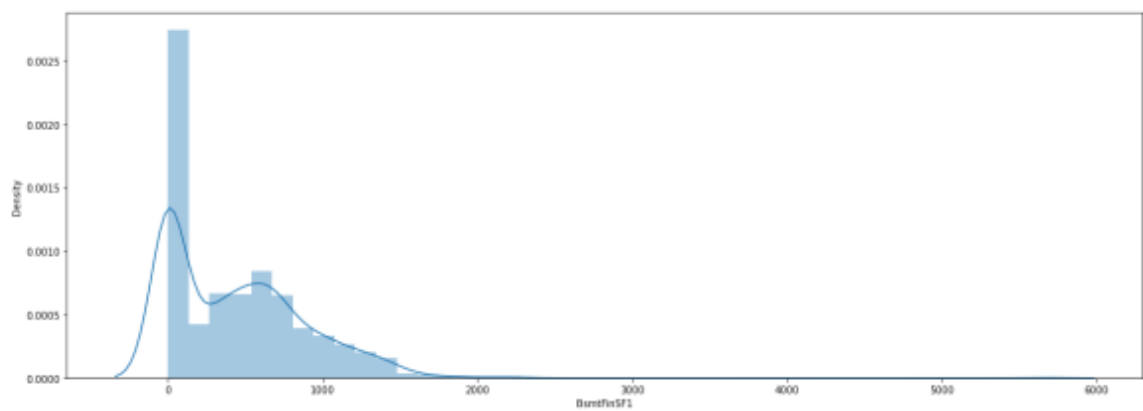
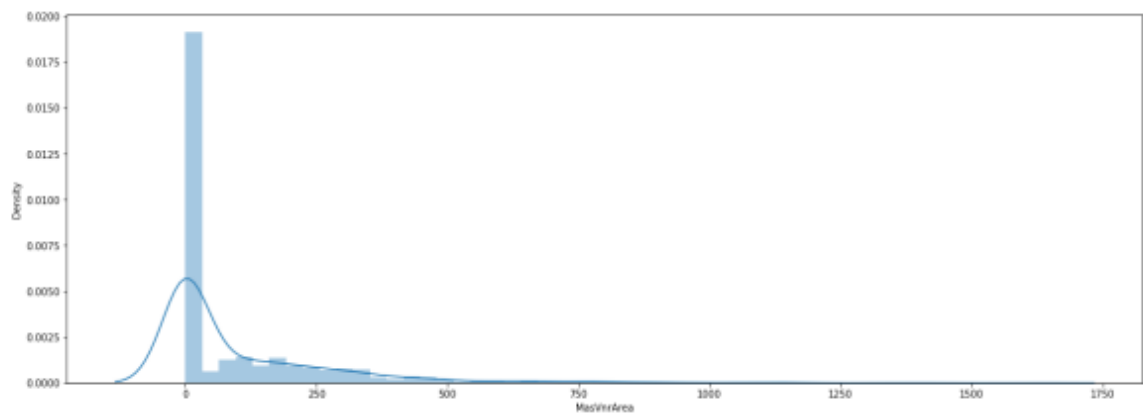
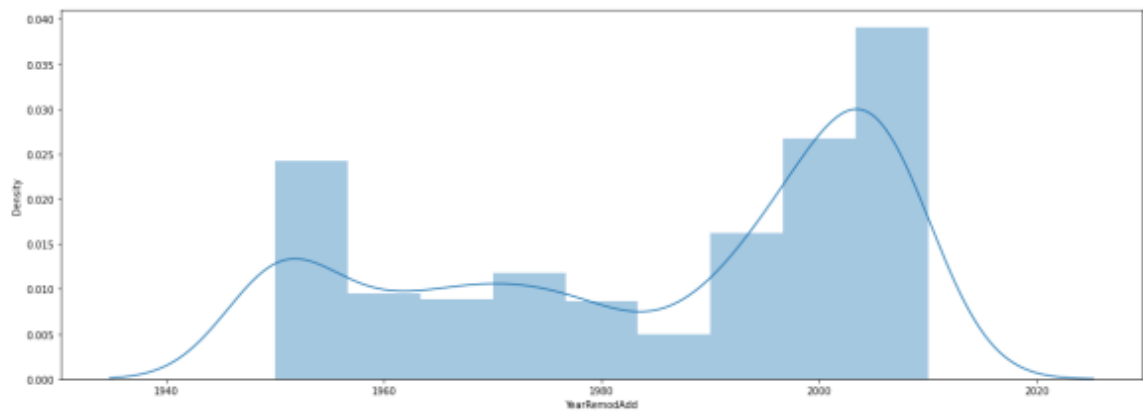
The same we have in classification problems which we call a threshold which is fixed at 0.5. So basically R^2 squared calculates how much regression line is better than a mean line. Hence, R^2 squared is also known as Coefficient of Determination or sometimes also known as Goodness of fit. R^2 Squared Now, how will you interpret the R^2 score? suppose If the R^2 score is zero then the above regression line by mean line is equal means 1 so $1-1$ is zero. So, in this case, both lines are overlapping means model performance is worst, It is not capable to take advantage of the output column. Now the second case is when the R^2 score is 1, it means when the division term is zero and it will happen when the regression line does not make any mistake, it is perfect. In the real world, it is not possible.

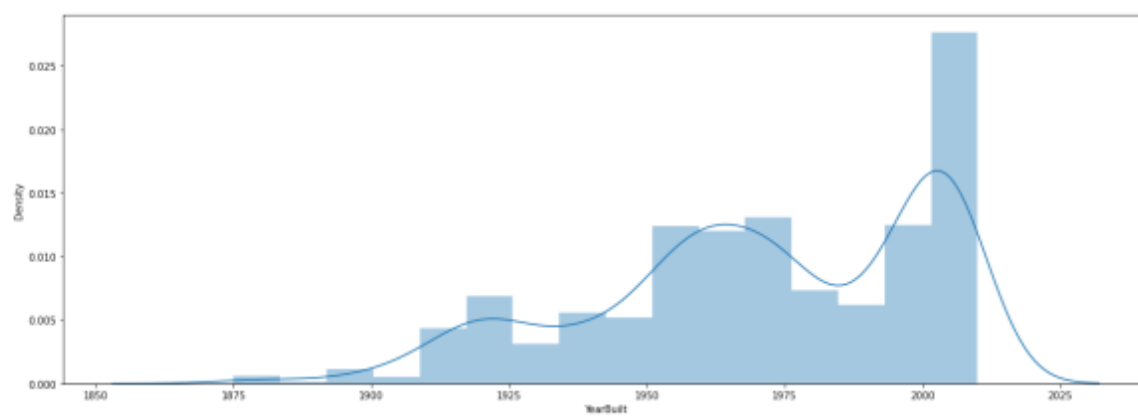
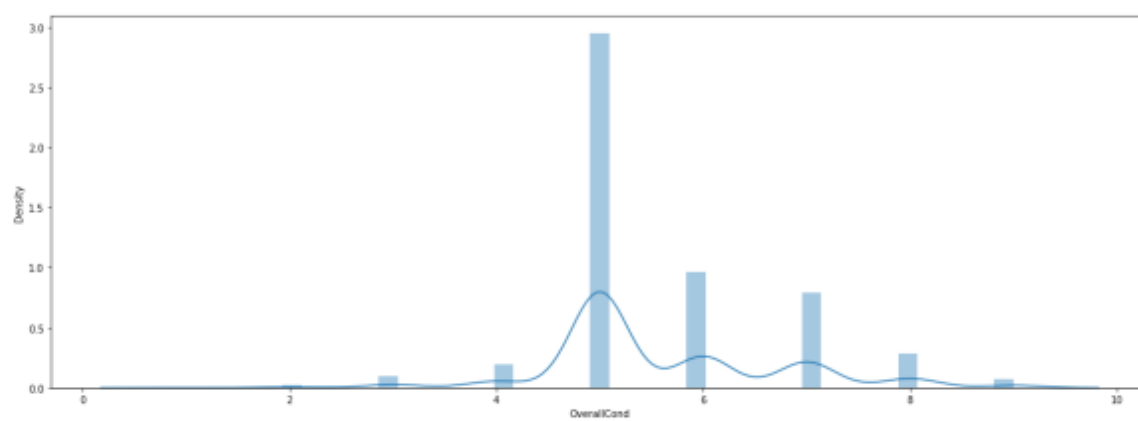
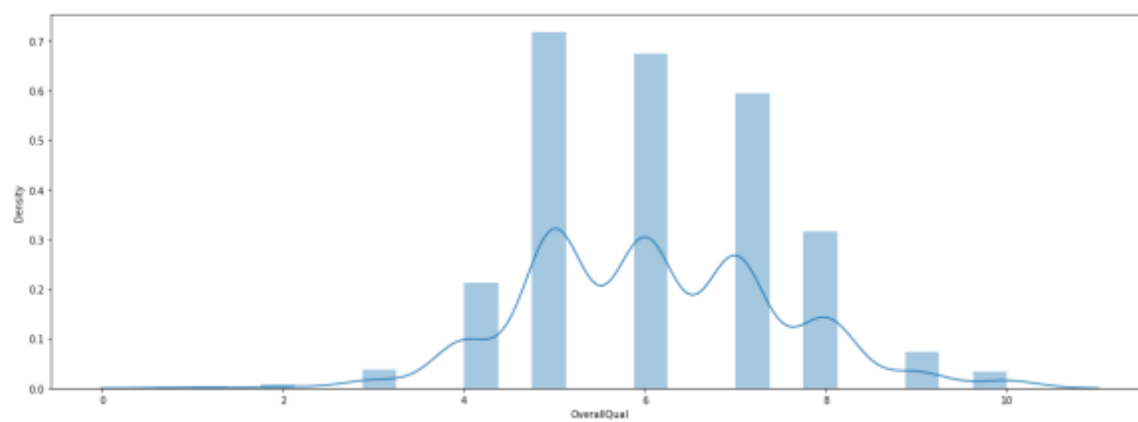
So we can conclude that as our regression line moves towards perfection, R^2 score move towards one. And the model performance improves. The normal case is when the R^2 score is between zero and one like 0.8 which means your model is capable to explain 80 per cent of the variance of data. `from sklearn.metrics import r2_score` $r2 = r2_score(y_test, y_pred)$.

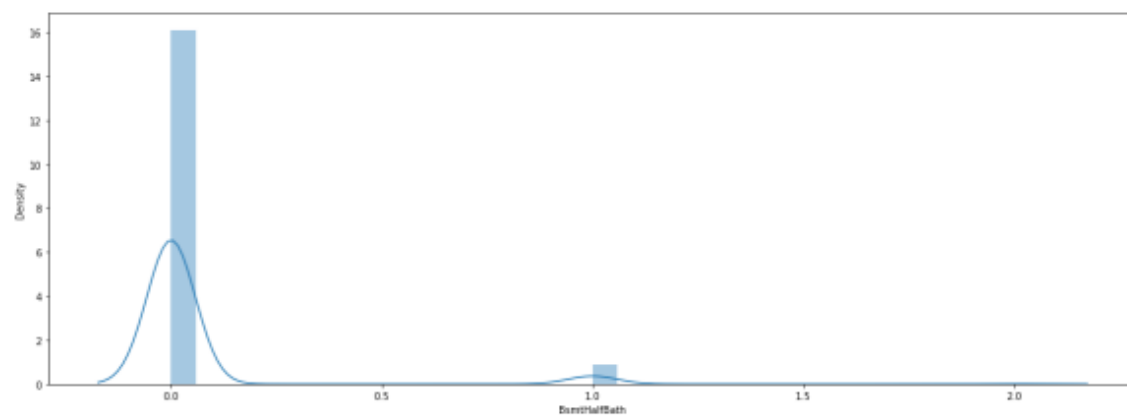
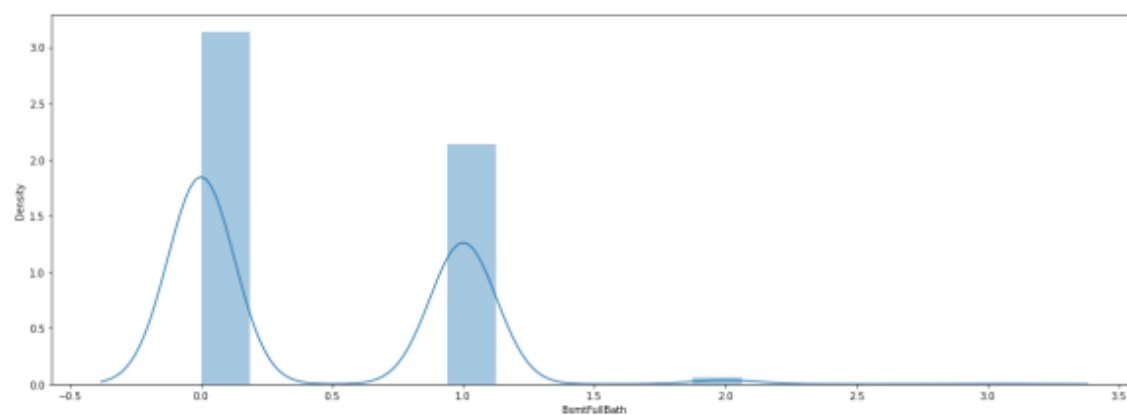
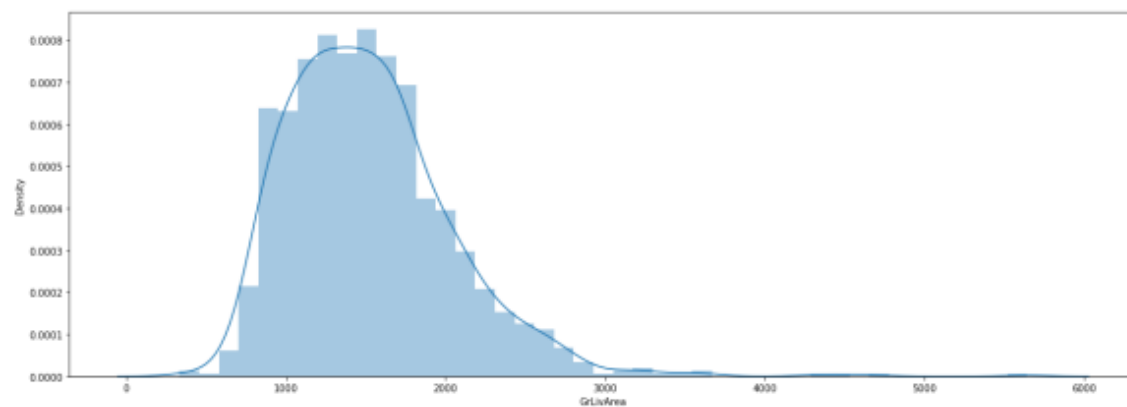
- Visualizations

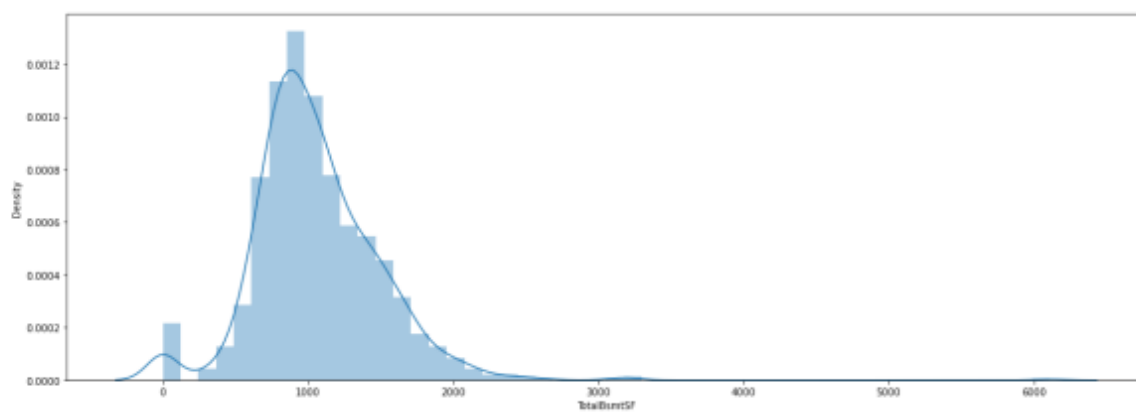
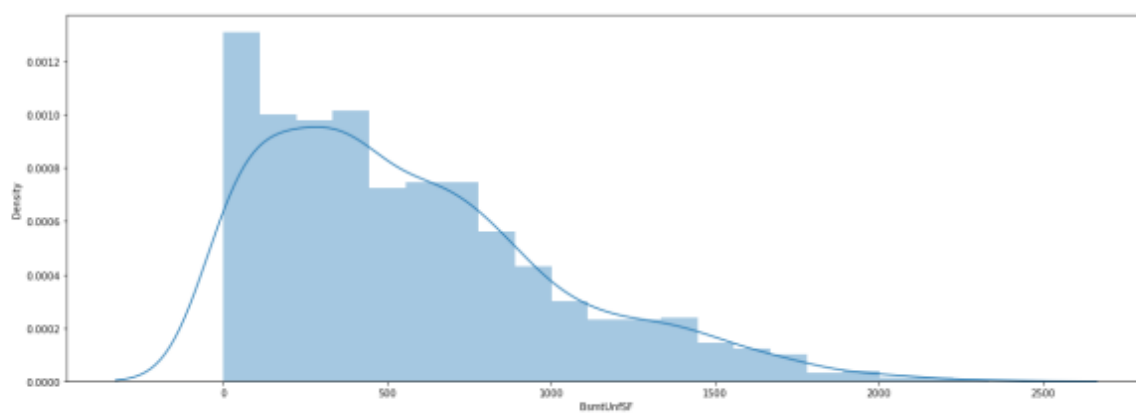
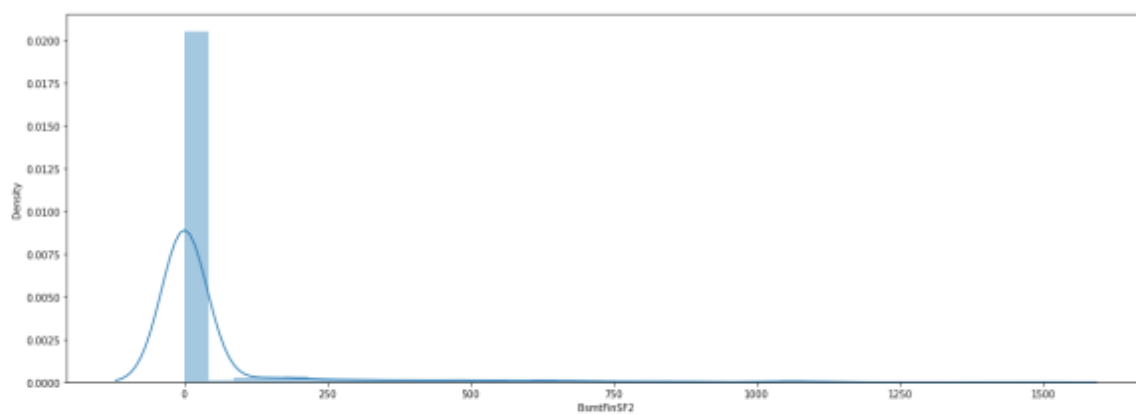
```
In [14]: counter=1;
for i in range(0,len(continous_columns)):
    plt.figure(figsize=(20,500))
    plt.subplot(60,1,counter)
    counter=counter+1
    sns.distplot(df[continous_columns[i]])
    plt.show()
```







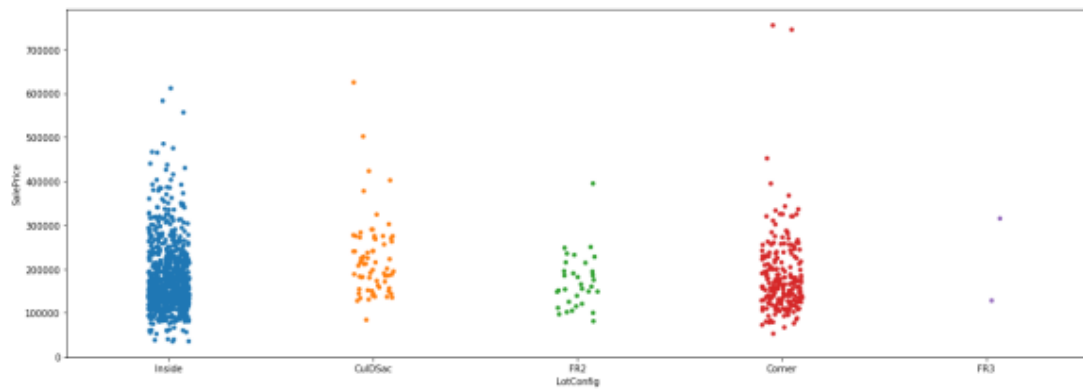
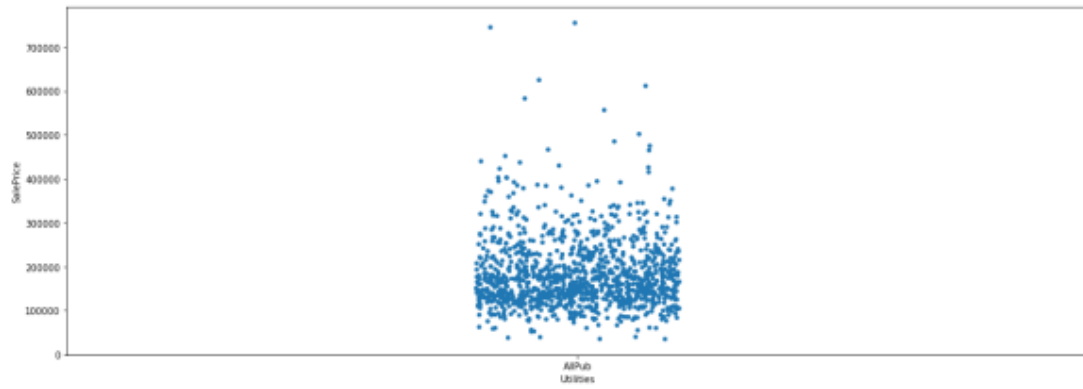
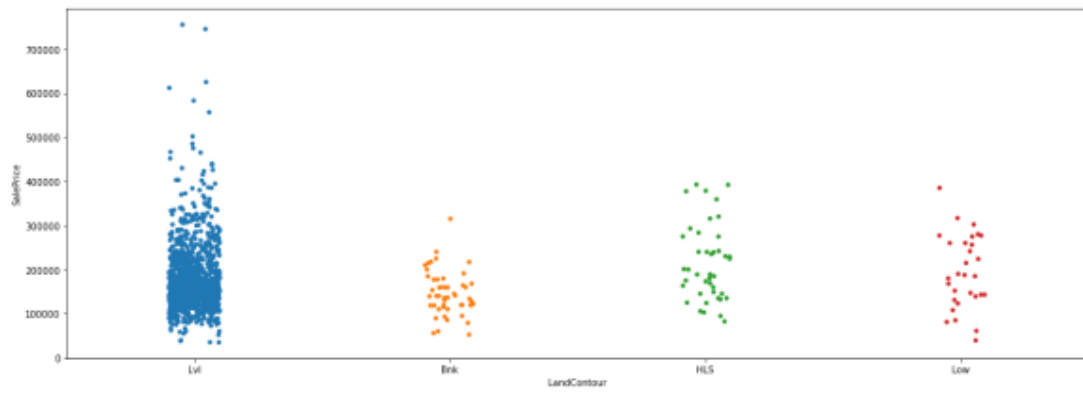
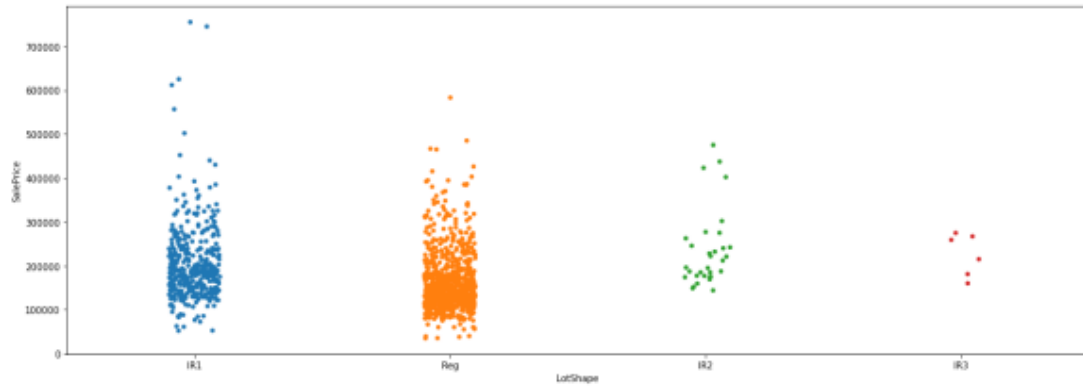


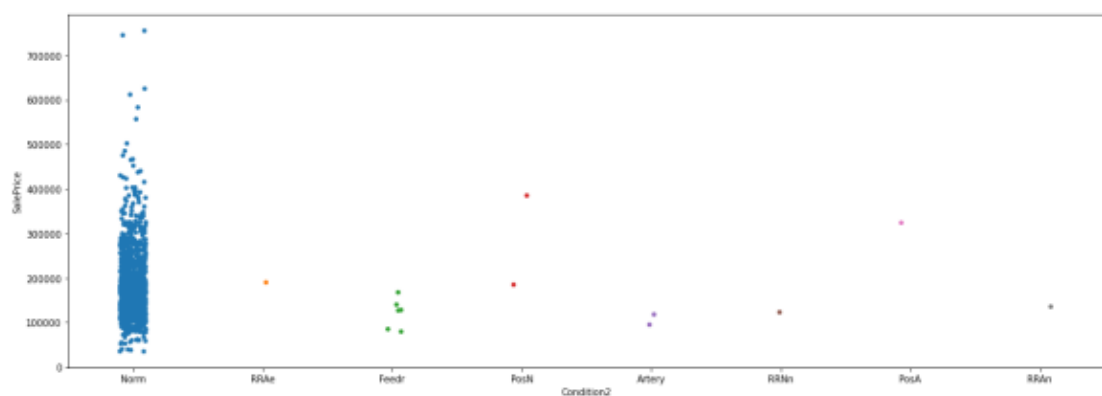
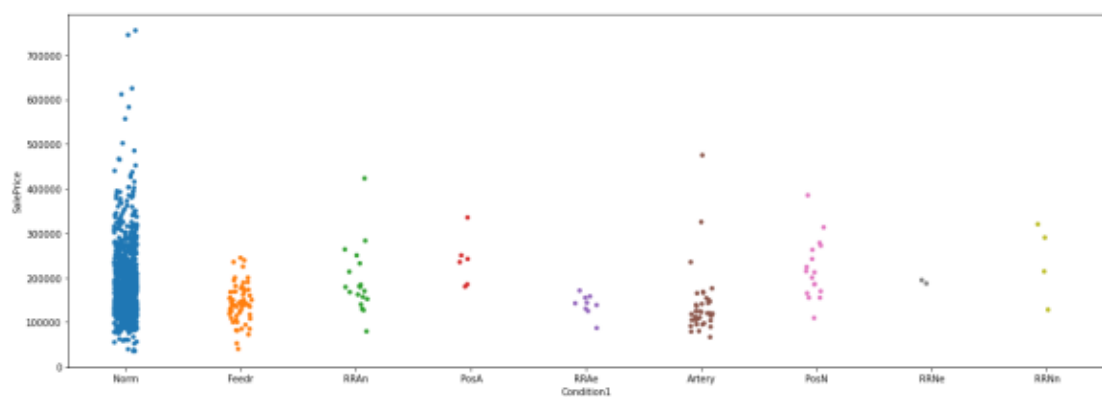
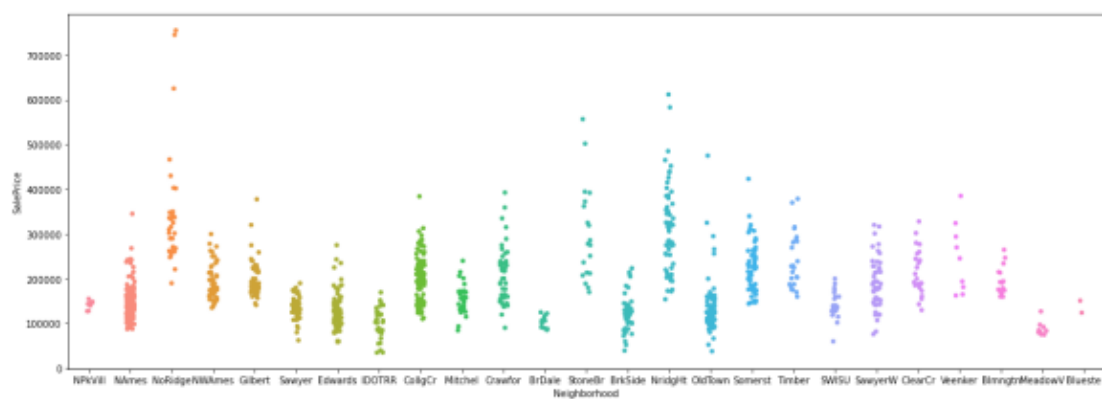
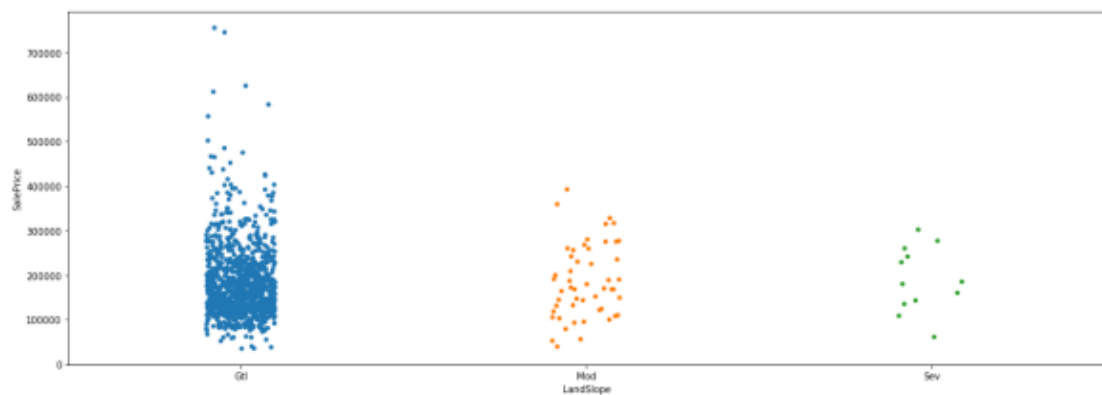


Findings:

MSSubClass -> not normally distributed
LotFrontage -> normally distributed
LotArea -> Normally distributed
OverallQual -> Not normally distributed
Overall cond -> Not normally distributed
Year Built -> Not normally distributed
Year remod add -> not normally distributed
BsmtFinSF1 -> not normally distributed
GarageBlt -> not normally distributed
Garage Area -> not normally distributed

```
In [15]: counter=1;
for i in range(0,len(continous_columns)):
    plt.figure(figsize=(20,500))
    plt.subplot(60,1,counter)
    counter=counter+1
    sns.boxplot(y=continous_columns[i],hue = continous_columns[i],data=df)
    #sns.boxplot(df[columns[i]])
    plt.show()
```



Findings:

MSZoning -> With RL zoning the property have higher value Street-> with Pave stype property have higher value LotShape-> IR1 shape property have higher value LandContour -> LVL property have higher value LotConfig -> Corner property have higher value LandSlope -> Gentle slope property have higher value Neighborhood -> NoRidge property have higher value Condition1 ->norm property have higher value Condition2 ->norm property have higher value BldgType -> 1Fam property have higher value HouseStyle -> 2 story property have higher value RoofStyle -> Gable and Hip stype property have higher value RoofMatl -> Compshg and WdShngle type property have higher value Exterior1st -> Brkcomm, Aspshnn style decreases the property value Exterior2nd -> Hd board type property have higher value MasVnrType -> BrkCmn type decreases the property value ExterQual -> Gd and Ex quality have higher property value Foundation -> Pconc foundation property have higher value BsmtQual -> Ex quality property have higher value BsmtCond -> Po quality property have low price BsmtFinType1 -> GLQ type have higher property prices BsmtFinType2 -> Unf type have higher price Heating -> GasA heating system have higher property price HeatingQC -> Houses with Fa HeatingQc price is low CentralAir -> Houses with central air have higher cost Electrical -> houses with FuseP and Mix have lower property value KitchenQual -> Excelent kitchen quality can increase the Property value GarageType -> Attached garage have higher property value GarageQual -> Poor garage quality decreases the price of property PavedDrive -> Paved drive hiuses have higher price SaleType -> WD and New sale type can get higher price SaleCondition -> having AdjLand have lower price

Interpretation of the Results Results

- : 1) Large amount of null values are present in the dataset
- 2) Data Set is not normally distributed
- 3) Dataset have outliers in most of the variables
- 4) Dataset is not normalized
- 5) Dataset is highly skewed
- 6) Random Forest Algorithm is best suited for the current datase